A Framework for Adaptive Service Guarantees*

Rajeev Agrawal
ECE Department
University of Wisconsin
Madison, WI 53706-1691
agrawal@engr.wisc.edu

R. L. Cruz
ECE Department
Univ. of Cal., San Diego
La Jolla, CA 92093-0407
cruz@ece.ucsd.edu

Clayton M. Okino
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755-8000
Clayton.M.Okino@Dartmouth.EDU

Rajendran Rajan
IBM T. J. Watson Research Center
Hawthorne, NY 10532
raju@watson.ibm.com

Abstract

Recently, a mathematical framework has been developed for the provision of deterministic quality of service guarantees in integrated services networks. This framework, or so-called "network calculus," involves the concepts of traffic envelopes, service curves, and convolution in the min-plus alegebra. Traffic envelopes constrain arrival processes, while service curves constrain the input-output behavior of network elements. Upper bounds on network delay are implied by the distance between a traffic envelope and service curve.

In this paper, we develop a somewhat parallel framework for the provision of deterministic quality of service guarantees to *adaptive* applications. Adaptive applications generate a traffic load that is dependent on network utilization, and thus characterizing the traffic generated from an adaptive application with an envelope is problematic. We introduce an adaptive service definition, through which upper bounds on network delay can be derived without using a traffic envelope. Instead, upper bounds on network delay are obtained in terms of the backlog and an *absolute* service curve. Since the backlog can be controlled through feedback, e.g. through window flow control, this yields a mechanism to obtain upper bounds on network delay. Lower bounds on network throughput are also implied through an absolute service curve.

1 Introduction

Adaptive network applications utilize feedback from the network to adjust the rate at which traffic is generated, so that network resources can be used efficiently. When an adaptive traffic source senses that the network is underutilized, it may choose to inject traffic into the network at a higher rate. The traffic may be generated from a fixed size data object, as in a file transfer, where the short term traffic rate is modulated

*Research supported in part by the National Science Foundation under grants NCR-93-05018, NCR 94-15684, NCR 97-25568, and by the Center for Wireless Communications at UCSD.

in accordance with a window flow control protocol. Alternatively, the total volume of traffic generated may in fact depend on the network feedback, as in adaptive multimedia compression algorithms that encode data with a lower distortion when the network is underutilized. With such adaptive applications, it is difficult to predict in advance the range of traffic behavior. At the same time, such adaptive applications may require quality of service guarantees from the network in order to function. For example, upper bounds on network latency may be required for interactive applications, and a network throughput guarantee may be required as well.

Recently, a mathematical framework has been developed for obtaining deterministic bounds on quality of service in integrated services networks [1][2][3][4][5][7]. Within this framework, upper bounds on delay are obtained with respect to a traffic envelope, which places an upper bound on the amount of data generated over intervals of time. For traffic that can be well characterized in terms of an envelope, bounds on delay, buffering requirements, and throughput can be obtained.

Adaptive applications, however, may generate traffic that is not well characterized by an envelope. Lower bounds on throughput, independent of a traffic envelope, are implied through a service curve [1]. Thus, within this mathematical framework, it is still possible to obtain throughput guarantees for adaptive applications. However, without a traffic envelope, it is difficult to obtain meaningful bounds on delay for an adaptive application that changes its traffic rate in response to network conditions.

In this paper, we develop a new mathematical framework which is useful for modelling services which guarantee upper bounds on delay *and* lower bounds on throughput, without requiring arrival traffic to be constrained by an envelope. We introduce an adaptive service definition which is the foundation of this framework. We shall see that the adaptive service definition is robust in the sense that we may obtain adaptive service guarantees for end-to-end systems that consist of network elements that themselves provide adaptive service guarantees. In particular, we shall obtain adaptive service guarantees for window flow control protocols. We shall also discuss a network element called an "elastic regulator" that can be used to allocate adaptive service guarantees, while allowing a traffic stream to utilize excess bandwidth.

In the next section, we shall review pertinent previous results and concepts, which provide the building blocks and the motivation for the mathematical framework developed in this paper.

2 Previous Results and Concepts

2.1 Notation and Preliminaries

We define a *process* to be a function of time, e.g. $A(t)$. Formally, A is a mapping from the real numbers into the extended non-negative real numbers, i.e. $A : \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}$. A process could count the amount of data arriving or departing to/from some network element, and in this case we may call the process an arrival process or a departure process, respectively. All processes are assumed to be non-decreasing and right continuous. We shall often consider what we call *causal* processes, which are simply processes which are identically zero for all negative times. For example, if A is a causal arrival process to a network element, then $A(t)$ is equal to the amount of data (in bits) arriving to the network element in the interval $(-\infty, t]$, and $A(t) = 0$ for all $t < 0$.

Given two processes A and B, the convolution of A and B is defined to be the function

$A * B : \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}$ such that

$$A * B(t) = \inf_{\tau \in \mathbb{R}} \{A(\tau) + B(t - \tau)\} \ .$$

It is easy to verify that $A * B$ is a process, i.e. it is non-decreasing, and right continuous. Furthermore, if A and B are causal, then $A * B$ is causal.

The convolution operator is **commutative** and **associative** , i.e. $A * B = B * A$ and $(A * B) * C = A * (B * C)$. Furthermore, if $B \wedge C$ denotes the pointwise minimum of B and C, then $A * (B \wedge C) = (A * B) \wedge (B * C)$. In other words, convolution is **distributive** with respect to the minimum operator. The identity element δ of this operator that satisifes $A * \delta = \delta * A = A$ may be verified to be

$$\delta(t) = \begin{cases} 0 & , t < 0 \\ \infty & , t \geq 0 \ . \end{cases}$$

Defining $\delta_d(t) := \delta(t - d)$, note that δ_d is a "shift element," i.e. for any process A we have $A * \delta_d(t) = A(t - d)$.

If G is a causal process, then[1] $G \leq \delta$. Thus, if F is any process and G is any causal process, then $F * G \leq F * \delta = F$.

The "deconvolution" of A and B, or A deconvolved with B, is defined by

$$A \oslash B(t) = \sup_{\tau \in \mathbb{R}} \{A(t + \tau) - B(\tau)\}$$

for all t. It is easy to verify that $A \oslash B$ is a process, i.e. it is non-decreasing, and right continuous. Furthermore, if A and B are causal, then $A \oslash B$ is causal. It can be verified that $A \oslash B$ is the smallest process H such that $H * B \geq A$. In other words, if $H * B \geq A$ then $H \geq A \oslash B$, and furthermore $(A \oslash B) * B \geq B$.

We use the notation x^+ to denote $\max\{x, 0\}$.

2.2 Service Guarantees

Suppose that the arrival of traffic to a network element is described by the cumulative arrival process R_{in}. The network element is a *service curve element* with *minimum service curve* S^{min} if the departure process R_{out} from the element satisfies $R_{out} \geq R_{in} * S^{min}$.

A process E is said to be an *envelope* for the process R if for all $\tau \leq t$ we have $R(t) - R(\tau) \leq E(t - \tau)$, or equivalently $R \leq R * E$.

A process E is said to be *sub-additive* if for all $t, \tau \in \mathbb{R}$ we have $E(\tau) + E(t - \tau) \geq E(t)$. Thus, if E is a sub-additive process, then $E * E \geq E$. It is usually assumed that envelopes are sub-additive and causal.

Given any process F, then we use the notation $F^{(\infty)}$ to denote the *sub-additive closure* [3] of F, i.e. $F^{(\infty)} = \wedge_{n=0}^{\infty} F^{(n)}$ and $F^{(n)}$ denotes the n-fold convolution with itself ($F^{(0)} = \delta$).

The backlog B of a network element describes the amount of data stored in the network element as a function of time. If the arrival and departure process of the network element are R_{in} and R_{out}, then the backlog $B(t)$ is given by

$$B(t) = R_{in}(t) - R_{out}(t) \ ,$$

[1]In this paper, all inequalities involving functions are defined in a pointwise sense.

assuming that no data is stored in the network element at time $t = 0$. If data from the network element departs in the same order that it arrives, then data arriving at time t will depart by time $t + \Delta$ only if $R_{out}(t + \Delta) \geq R_{in}(t)$. This motivates the following definition of $d(t)$, the virtual delay at time t:

$$d(t) = \inf\{\Delta : \Delta \geq 0 \text{ and } R_{out}(t + \Delta) \geq R_{in}(t)\} \ .$$

One of the fundamental previous results concerning envelopes and service curves is that delay can be upper bounded by maximum horizontal distance between an envelope and a service curve. This is formalized in the following proposition.

Proposition 1 *[5][1][7][2] Consider a service curve element with minimum service curve S^{min}. If the arrival process has envelope E, then the virtual delay $d(t)$ satisfies $d(t) \leq d^{max}$ for all t, where*

$$d^{max} = \inf\{d : d \geq 0 \ and \ E * \delta_d \leq S^{min}\} \ .$$

It is easy to see that if the arrival process to a service curve element is unconstrained, then the virtual delay can be unbounded, since the backlog may be unbounded. Adaptive applications may use feedback to adjust the arrival process to a network so that the backlog remains bounded. However, even if the backlog of a service curve element remains bounded, the virtual delay may still be unbounded. This is because the condition $R_{out} \geq R_{in} * S^{min}$ is not strong enough to guarantee that R_{out} will increase over an arbitrary interval where the backlog is positive. For example, suppose $R_{out}(s)$ is significantly larger than $R_{in} * S^{min}(s)$, say $R_{out}(s) \geq R_{in} * S^{min}(s + \Delta)$, where Δ is large. In this case, the constraint $R_{out}(u) \geq R_{in} * S^{min}(u)$ can be met for $u \in (s, s + \Delta)$ even when no data departs the network element in the interval $[s, s + \Delta]$, i.e. $R_{out}(s + \Delta) = R_{out}(s)$. In other words, roughly speaking, if a traffic stream receives significantly more service than it is guaranteed up to a time s, then after time s it is possible for a service curve element to serve very little traffic, and still meet the guarantee specified by the service curve. In order to guarantee an adaptive session a bounded delay, it should not be "punished" later for utilizing excess bandwidth at an earlier time. This motivates the adaptive service definition introduced in the next section.

3 Adaptive Service Guarantees

Definition 2 (Adaptive Service Guarantee) *Given a network element with arrival process R_{in} and departure process R_{out}. Let $\tilde{S}$ and S be causal processes. We say the network element adaptively guarantees $(S, \tilde{S})$ over $[s^*, t^*]$, if for all $s, t \in [s^*, t^*]$ and $s \leq t$ we have*

$$R_{out}(t) \geq \{R_{out}(s) + \tilde{S}(t - s)\} \wedge \inf_{u: t \geq u \geq s} \{R_{in}(u) + S(t - u)\} \ . \qquad (1)$$

If (1) holds for all $s, t \in [s^*, t^*]$ and $s \leq t$, then we write, as a shorthand notation,

$$R_{in} {\rightarrow} (S, \tilde{S})_{[s^*, t^*]} {\rightarrow} R_{out} \ .$$

If (1) holds for all $s, t \in \mathbb{R}$ and $s \leq t$, then we simply say that the network element adaptively guarantees $(S, \tilde{S})$, and we write as a shorthand notation

$$R_{in} \to (S, \tilde{S}) \to R_{out} \ .$$

In the context of these definitions, we say that S is a partial service curve, and that $\tilde{S}$ is an absolute service curve.

In the special case where $S = \tilde{S}$ and $R_{in} \to (S, \tilde{S})_{[s^*,t^*]} \to R_{out}$, we simply say that the network element adaptively guarantees $\tilde{S}$ over $[s^*, t^*]$, and we write $R_{in} \to (\tilde{S})_{[s^*,t^*]} \to R_{out}$. Similarly, if $R_{in} \to (\tilde{S}, \tilde{S}) \to R_{out}$ then we simply say that the network element adaptively guarantees $\tilde{S}$, and we write $R_{in} \to (\tilde{S}) \to R_{out}$.

To get a rough sense of the meaning of these definitions, suppose that $R_{in} \to (S, \tilde{S}) \to R_{out}$, and fix $s \leq t$. Note that the lower bound on $R_{out}(t)$ in (1) potentially increases with $R_{out}(s)$, and it is in this sense that the guarantee is adaptive. Said another way, the amount traffic that departs over (s,t), namely $R_{out}(t) - R_{out}(s)$ is at least $\tilde{S}(t-s)$, assuming that R_{in} is sufficiently large over (s,t), and this is independent of how much traffic departed prior to time s. This also motivates why we have called $\tilde{S}$ an absolute service curve. The partial service curve S influences the lower bound on $R_{out}(t) - R_{out}(s)$ in the case where R_{in} is "small" over the interval (s,t). Other motivations for the terminology will become more clear once we examine systems composed of network elements that provide adaptive service guarantees.

Next, we give two simple examples of network elements which provide adaptive guarantees.

Example 3 (Constant Delay Element) *Consider a network element with constant delay $d \geq 0$, where $R_{out}(t) = R_{in}(t-d)$ for all t. There holds $R_{in} \to (\delta_d) \to R_{out}$.*

Example 4 (GPS Server) *Consider a session passing through a Generalized Processor Sharing (GPS) [9] server, with arrival process R_{in} and departure process R_{out}. One of the properties of a GPS server is that each session passing through it has a guaranteed minimum bandwidth, say ϕ. In particular, if $B(u) = R_{in}(u) - R_{out}(u) > 0$ for $u \in (s,t)$, then $R_{out}(t) - R_{out}(s) \geq \phi(t-s)$. Define the process μ_ϕ according to $\mu_\phi(t) = \phi t$ if $t \geq 0$ and $\mu_\phi(t) = 0$ for $t < 0$. It is not difficult to show that the GPS server adaptively guarantees μ_ϕ to the session, i.e. $R_{in} \to (\mu_\phi) \to R_{out}$.*

The guarantee provided by GPS motivates the following definition.

Definition 5 (Strong Service Guarantee) *Consider a network element with arrival process R_{in}, departure process R_{out}, and backlog $B = R_{in} - R_{out}$. Let S^* be a process. Suppose the network element has the property that for all intervals (s,t) where $\inf\{B(u) : u \in [s,t]\} > 0$ there holds $R_{out}(t) - R_{out}(s) \geq S^*(t-s)$. In this case, we say that the network element strongly guarantees S^*, and we call S^* a strong service curve.*

For example, one of the defining properties of a GPS server is that it strongly guarantees μ_ϕ, in the context of Example 4. In fact, a strong service curve is consistent with the notion of the "universal service curve" originally introduced in [9]. An adaptive service guarantee implies a strong service guarantee, as we now discuss.

Proposition 6 (Strong Service from Adaptive Service Guarantee) *Consider a network element with arrival process R_{in} and departure process R_{out}. If $R_{in} \to (S, \tilde{S}) \to R_{out}$, then the network element strongly guarantees S^*, where $S^* = \tilde{S} * S^{(\infty)}$.*

For example, since $\mu_\phi = \mu_\phi * \mu_\phi$, it follows that if a network element adaptively guarantees μ_ϕ, then it stongly guarantees μ_ϕ.

As another example, consider a network element with constant delay $d > 0$. Combining the results of Example 3 and Proposition 6, it follows that a constant delay element strongly guarantees $S^* \equiv 0$. It is clear that, while trivial, this is the best result possible since we may inject an arbitrarily small amount of traffic into the delay element for an arbitrarily long interval of time, while keeping the backlog strictly positive throughout the interval.

The following result illustrates a distinction between an strong service guarantee and an adaptive service guarantee.

Proposition 7 *If $R_{in} \to (S, \tilde{S}) \to R_{out}$, and the network element has a backlog at time $s \le t$ such that $B(s) \ge \tilde{S}(t-s)$, then $R_{out}(t) - R_{out}(s) \ge \tilde{S}(t-s)$.*

For all x, define the "pseudo inverse" of $\tilde{S}$, $\tilde{S}^{-1}$ according to

$$\tilde{S}^{-1}(x) = \inf\{y : y \ge 0, \tilde{S}(y) \ge x\} \ .$$

Using the right continuity of $\tilde{S}$, it follows that $\tilde{S}(\tilde{S}^{-1}(x)) \ge x$ for all x.

We now present an upper bound on virtual delay in terms of the backlog, which is almost equivalent to Proposition 7.

Proposition 8 (Delay Bound from Backlog) *If $R_{in} \to (S, \tilde{S}) \to R_{out}$, then the virtual delay at time t, $d(t)$, is upper bounded according to $d(t) \le \tilde{S}^{-1}(B(t))$.*

The significance of Proposition 8 is that the delay bound does not require the arrival process R_{in} to be constrained by an envelope, but only that the backlog B is bounded. One possible way to ensure that the backlog is bounded is to employ feedback, as we shall discuss later, within the context of window flow control.

3.1 Composition Results

We first consider a tandem series of n network elements, with no feedback.

Proposition 9 (Composition of Adaptive Service Guarantees) *Suppose for $i = 1, \cdots, n$, we have*

$$R_{i-1} \to (S_i, \tilde{S}_i)_{[s^*,t^*]} \to R_i \ .$$

Then

$$R_0 \to (S_1 * S_2 * \cdots * S_n, \tilde{G})_{[s^*,t^*]} \to R_n \quad ,$$

where $\tilde{G} = (\tilde{S}_1 * S_2 * \cdots * S_n) \wedge (\tilde{S}_2 * S_3 * \cdots * S_n) \wedge \cdots \wedge (\tilde{S}_{n-1} * S_n) \wedge \tilde{S}_n$.

Next, we consider network elements with feedback.

Consider the closed loop window based flow control model depicted in Figure 1, which consists of network element N_1, network element N_2, and another network element called a *throttle*. The arrival process to the throttle is R_{in}, and the departure process of the throttle, $\hat{R}$, is as large as possible, subject to a "throttle control process" $W + F$. In particular, we have

$$\hat{R}(t) \quad = \quad R_{in}(t) \wedge \{F(t) + W(t)\} \ ,$$

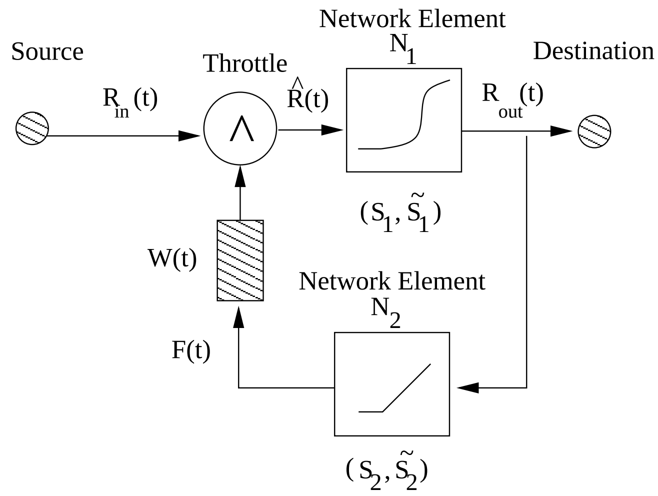

Figure 1: A single hop with window flow control

for all t. Thus, the throttle may store traffic. The throttle departure process feeds network element N_1, whose departure process is denoted as R_{out}. Finally, the arrival process to network element N_2 is R_{out}, and the departure process of N_2 is given by the process F.

We call W the *window size process*, even though it may be a decreasing function. It is assumed, however, that the throttle control process $F + W$ is a process, i.e. it is a non-negative, non-decreasing, right-continuous function.

Note that the total amount of traffic stored in N_1 and N_2 at time t is given by $\hat{R}(t) - F(t) \leq (F(t) + W(t)) - F(t) = W(t)$, and thus $W(t)$ controls the backlog in N_1 and N_2.

Proposition 10 (Adaptive Service Guarantee for Window Flow Control) *If* $\hat{R} \rightarrow (S_1, \tilde{S}_1)_{[s^*,t^*]} \rightarrow R_{out}$, $R_{out} \rightarrow (S_2, \tilde{S}_2)_{[s^*,t^*]} \rightarrow F$, $\hat{R} = R_{in} \wedge (F + W)$, *and* $W(u) = w > 0$ *for all* $u \in [s^*, t^*]$, *then* $R_{in} \rightarrow (Y, \tilde{Y})_{[s^*,t^*]} \rightarrow R_{out}$, *where* $Y = \wedge_{n=0}^{\infty}[S_1 * G^{(n)} + nw]$, $\tilde{Y} = \wedge_{n=0}^{\infty}[\tilde{G} * G^{(n)} + nw]$, $G = S_1 * S_2$, *and* $\tilde{G} = \tilde{S}_1 \wedge (\tilde{S}_2 * S_1)$.

Corollary 11 *Suppose* $\hat{R} \rightarrow (S_1, \tilde{S}_1)_{[s^*,t^*]} \rightarrow R_{out}$, $R_{out} \rightarrow (S_2, \tilde{S}_2)_{[s^*,t^*]} \rightarrow F$, $\hat{R} = R_{in} \wedge (F + W)$, *and* $W(u) = w > 0$ *for all* $u \in [s^*, t^*]$. *If* $w \geq \sup_{t \in \mathbb{R}}\{\tilde{G}(t) - (\tilde{G} * G)(t)\}$ *and* $w \geq \sup_{t \in \mathbb{R}}\{S_1(t) - (S_1 * G)(t)\}$, *then* $R_{in} \rightarrow (S_1, \tilde{G})_{[s^*,t^*]} \rightarrow R_{out}$.

Example 12 *Consider network illustrated in Figure 1, in the special case where* N_1 *represents the tandem configuration of a local GPS server with bandwidth guarantee* ϕ, *a constant propagation delay of* d_F, *and a remote GPS server with bandwidth guarantee* ϕ. *Also, suppose* N_2 *represents constant propagation delay of* d_R. *In this case,* N_1 *adaptively guarantees* $\tilde{S}_1 = \mu_\phi * \delta_{d_F}$, *and* N_2 *adaptively guarantees* δ_{d_R}. *Corollary 11 implies that if* w *is equal to the bandwidth delay product, i.e.* $w = \phi(d_F + d_R)$, *then* $R_{in} \rightarrow (\mu_\phi * \delta_{d_F}, \mu_\phi * \delta_{d_F} * \delta_{d_R})_{[s^*,t^*]} \rightarrow R_{out}$. *Furthermore, suppose that the source adapts to the feedback, such that the backlog in the throttle does not exceed* b_T^{max}. *In this case, since the backlog in network element* N_1 *is upper bounded by* $w = \phi(d_F + d_R)$, *it follows that* $R_{in} - R_{out}$ *is upper bounded by* $w + b_T^{max}$. *Hence, applying Proposition 8, it follows that the network delay is upper bounded by* $2(d_F + d_R) + b_T^{max}/\phi$.

Next, we consider the issue of how to construct network elements that provide adaptive service guarantees. We begin with a motivating proposition.

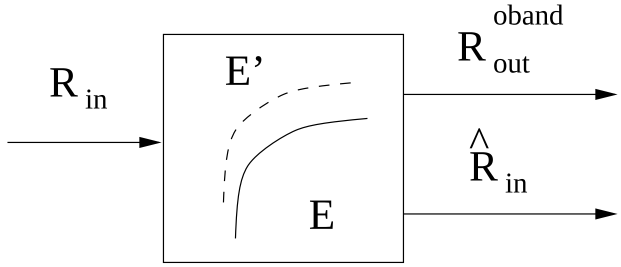

Figure 2: An Elastic Regulator

Proposition 13 (Basic Synthesis of Adaptive Service Guarantee) *Consider a network element that guarantees minimum service curve S^{min}, and suppose the arrival process is known to have the traffic envelope E. Then the network element adaptively guarantees $(S, \tilde{S})$, where $S = S^{min}$ and*

$$\tilde{S}(x) = \inf_{y:y\geq 0}\{\left[S^{min}(x+y) - E(y)\right]^{+}\} \quad \textit{for all } x.$$

Proposition 13 suggests a way to synthesize a network element that provides an adaptive guarantee. First, we "shape" traffic with a regulator [1][3] so that it has an envelope E. The departure process of the regulator thus has envelope E, and is then fed to a minimum service curve element with minimum service curve S^{min}. According to Proposition 13, the system then adaptively guarantee $(S^{min}, \tilde{S})$. Note that in order to maximize $\tilde{S}$, we wish to set E as small as possible. However, if E is too small, then the minimum service curve of the entire system $E * S^{min}$, will be smaller than S^{min}. This then motivates choosing $E \geq S^{min} \oslash S^{min}$. A problem with this approach is that the regulator may prevent utilization of excess bandwidth that may be offered.

3.2 Elastic Regulator

This motivates us to consider a new network element, called an "elastic regulator." which has a single arrival process, R_{in} and a departure process that is decomposed into two sub-departure processes, corresponding to a "conformant" departure process, $\hat{R}_{in}$, and an "out-of-band" departure process, R_{out}^{oband}, as illustrated in Figure 2. The elastic regulator ensures, essentially, that $\hat{R}_{in}$ has envelope E, and thus we refer to packets that are counted in $\hat{R}_{in}$ as conformant packets. In order to utilize excess bandwidth, packets may depart "out of band." Such packets are counted in the "out-of-band" departure process R_{out}^{oband}, and are called out-of-band packets.

We shall define the elastic regulator with respect to an arrival process of the form

$$R_{in}(t) = \sum_{k=1}^{\infty} L^k u(t - \tau^k) \ ,$$

where $u(x) = 1$ if $x \geq 0$ and $u(x) = 0$ if $x < 0$. In particular, packets arrive instantaneously at the instants $\tau^0 \leq \tau^1 \leq \tau^2 \cdots$, and L^k denotes the length in bits of the k^{th} arriving packet. The backlog of the elastic regulator is given by $B^r(t) = R_{in}(t) - \hat{R}_{in}(t) - R_{out}^{oband}(t)$. Packets may depart "out-of-band" at time t if $B^r(t) > 0$ or if $t = \tau^k$ for some k. We do not otherwise specify when packets may depart out of band, and in fact this is determined by a downstream network element. The elastic regulator is defined by specifying when conformant packets may leave.

Toward this end, suppose that E is a causal, sub-additive envelope. We shall in fact also assume that $E(t)$ is *left*-continuous for $t > 0$, which does not appear to be a significant loss of generality. We assume that there is a maximum packet size L_{max}, i.e. $\sup\{L^k : k \geq 0\} \leq L_{max}$.

We represent the conformant departure process as

$$\hat{R}_{in}(t) = \sum_{k=0}^{\infty} \hat{L}^k u(t - \hat{\tau}^k) ,$$

where $\hat{L}^k$ is the number of bits of the k^{th} conforming packet that departs, and we define $\hat{\tau}^0 = 0$ and $\hat{L}^0 = 0$. We also assume $0 \leq \hat{\tau}^k \leq \hat{\tau}^{k+1}$ for all k.

We define the envelope E' as follows:

$$E'(x) \quad = \quad \begin{cases} E(x) + L_{max} & \text{, for } x \geq 0 \\ 0 & \text{elsewhere} \end{cases} .$$

We will refer to E as the "target" envelope, since the $\hat{R}_{in}$ will not in general have envelope E, but only have envelope E'.

We describe when conforming packets may depart in terms of a sequence $\{\tau_k'\}$, defined below. We call τ_k' the eligibility time of the k^{th} conformant departure. This eligibility time is based on *previous* conformant departures from the elastic regulator and is recalculated each time *any* conformant departure takes place. The sequence $\{\tau_k'\}$ is defined as follows. Let $\tau_1' := 0$, $\hat{\tau}^0 := 0$, and then τ_{k+1}' is computed recursively from $(\hat{\tau}^1, \hat{\tau}^2, \cdots, \hat{\tau}^k)$ at time $\hat{\tau}^k$. In particular

$$\tau_{k+1}' = \inf\{s : s \geq \hat{\tau}^k \text{ and } \min_{j:1 \leq j \leq k} [\sum_{l=0}^{j-1} \hat{L}^l + E(s - \hat{\tau}^j)] \geq \sum_{l=0}^{k} \hat{L}^l\} .$$

By definition, the k^{th} conformant departure can be no earlier than τ_k'. The k^{th} confomant departure is at time τ_k', unless $\lim_{t \to (\tau_k')^-} (B_i^r(t)) = 0$ and there is no arrival to the elastic regulator at time τ_k'. In this case, the time of the k^{th} conformant departure is the time of the first arrival to the elastic regulator after time τ_k'.

Lemma 14 *An elastic regulator with target envelope E has the following properties:*

(*i*) *The conformant departure process has envelope E', i.e. $\hat{R}_{in} \leq \hat{R}_{in} * E'$.*

(*ii*) *For $B^r(w) > 0$, there exists $x < w$ such that $\hat{R}_{in}(w) - \hat{R}_{in}(x) > E(w - x)$.*

(*iii*) *Given u' and any $\epsilon > 0$, there exists $w < u' + \epsilon$ and there exists $w^* \in (w - \epsilon, w + \epsilon)$ such that*

$$\hat{R}_{in} * E(u') \geq \hat{R}_{in}(w) + E(u' - w) - \epsilon ,$$

$B^r(w^*) = 0$, and $\hat{R}_{in}(w) = \hat{R}_{in}(w^*)$.

Proposition 15 *Consider a system composed of an elastic regulator with target envelope E and a service curve element with minimum service curve S^{min}. The arrival process of the system directly feeds the elastic regulator. The conformant departures of the elastic regulator feed the service curve element. Finally, the departure process of the system, R_{out}, is given $R_{out} = \hat{R}_{out} + R_{out}^{oband}$, where $\hat{R}_{out}$ is the departure process of the service* *curve element and R^{oband}_{out} is the out-of-band departure process from the elastic regulator. If $E \geq S^{min} \oslash S^{min}$, there holds $R_{in} \rightarrow (S, \tilde{S}) \rightarrow R_{out}$, where $S = S^{min}$ and*

$$\tilde{S}(x) = \inf_{y: y \geq 0} \{ \left[S^{min}(x+y) - E'(y) \right]^+ \} \quad \textit{for all } x.$$

The significance of Proposition 15 is that out of band departures from the elastic regulator allow excess bandwidth to be utilized, but do not jeopardize the benefit of the adaptive service guarantee provided by the minimum service curve element, as implied by Proposition 13. In particular, similar to [6], elastic regulators may be employed at the input of a scheduler that provides prescribed minimum service curve guarantees, as in the "SCED" scheduler in [5][10]. In this context, out-of-band packets are served when the scheduler backlog is zero. This will result in a system that allows excess bandwidth to be utilized, while providing prescribed adaptive service guarantees. Details can be found in [8]

References

[1] R. Agrawal, R. L. Cruz, C. Okino, R. Rajan, Performance bounds for flow control protocols. Technical Report ECE-98-1, ECE Dept., University of Wisconsin - Madison, May 1998.

[2] R. Agrawal and R. Rajan. A general framework for analyzing schedulers and regulators. *Proceedings of the 34th Annual Allerton Conference on Communication, Control, and Computing,* Monticello, Ill., Oct. 1996, pp. 239-248.

[3] C.-S. Chang. On deterministic traffic regulation and service guarantees: a systematic approach by filtering. *IEEE Transactions on Information Theory,* vol. 44, no. 3, May, 1998, pp. 1097-1110.

[4] R.L. Cruz. A calculus for network delay, Part I: Network Elements in Isolation, *IEEE Trans. on Information Theory,* vol. 37, no. 1, Jan. 1991, pp. 114-131.

[5] R.L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, 1995, pp. 1048-56.

[6] L. Georgiadis, R. Guérin, V. Peris, and K.N. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking,*vol. 4, no. 4, August, 1996, pp. 482-501.

[7] J.-Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory,* vol. 44, no. 3, May 1998, 1087-1096.

[8] Clayton M. Okino, A framework for performance guarantees in adaptive communication networks. Ph.D. Thesis, Dept. of Electrical and Computer Engineering, UCSD, September 1998.

[9] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, June 1993, pp. 344-57.

[10] H. Sariowan, R. L. Cruz, and G. C. Polyzos. Scheduling for quality of service guarantees via service curves. *Proc. International Conference on Computer Communications and Networks '95 (ICCCN'95)*, Sept. 1995, pp. 512-20.